

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА БАЗЕ ОПЕРАЦИОННОЙ СИСТЕМЫ ANDROID ДЛЯ УПРАВЛЕНИЯ НЕСКОЛЬКИМИ УСТРОЙСТВАМИ

¹Арсентьев Е.А., ¹Бабич М.В., ¹Шемякин Д.Г., ¹Фрелих А.В.

¹ ФГАОУ ВПО «Уральский Федеральный Университет имени первого Президента России Б. Н. Ельцина», Екатеринбург, Россия (Россия, г.Екатеринбург, ул.Мира, 19), e-mail: arsentiev.evgeny@gmail.com

Аннотация: В настоящей статье рассматривается схема взаимодействия «Оператор медицинского устройства – медицинское устройство – пациент» в лечебном процессе, когда врачу необходимо настраивать прибор непосредственно в течение данного процесса. Представлены преимущества разбиения блока «медицинское устройство» на два блока – «пульт управления» и «исполняющее устройство». Для пульта управления медицинскими устройствами было разработано программное обеспечение, на базе операционной системы Android, позволяющее манипулировать параметрами медицинских приборов. Отличительной особенностью разработанного приложения является возможность работать с несколькими устройствами одновременно, что позволяет изменять их параметры быстрее, без наличия пультов управления персонально для каждого прибора. Рассматривается алгоритм функционирования и пример интерфейса данного программного решения.

Ключевые слова: Android, медицинское устройство, программное обеспечение, манипуляция параметрами, пульт управления.

DEVELOPMENT OF SOFTWARE BASED ON OPERATING SYSTEM ANDROID FOR MANAGEMENT OF MULTIPLE DEVICES

¹Arsentiev E.A., ¹Babich M.V., ¹Shemyakin D.G., ¹Frelich A.V.

¹The Ural Federal University named after the first President of Russia B. N. Yeltsin, Yekaterinburg, Russia (Russia, Yekaterinburg, street Mira, 19), arsentiev.evgeny@gmail.com

Abstract: The article considers the scheme of interaction «operator of the medical device – medical device – patient» in medical process, when the doctor must configure the device during this process. The advantages of separation of block «medical device» into two blocks «control panel» and «executing device» are evaluated. We develop software based on operating system Android for control panel, which allows manipulate parameters of medical devices. The distinctive feature of this application is possibility to control several devices at the same time, that allows configure their parameters faster without separate control panel for each device. In this article we consider an algorithm of the functioning and an example of the interface for control panel.

Key words: Android, medical device, software, manipulation of parameters, control panel.

При лечении пациентов медицинскими физиотерапевтическими устройствами, зачастую существует механический контакт между этим прибором и самим пациентом. В то же время, требуется постоянный мониторинг лечебного процесса со стороны врача. Схема такого взаимодействия представлена на рисунке 1. Данные обстоятельства приводят к проблемам, таким как нахождение врача в физическом поле, генерируемом медицинским

устройством, и ограничение мобильности пациента с таким прибором. Всё это может отрицательно влиять на лечебный процесс.

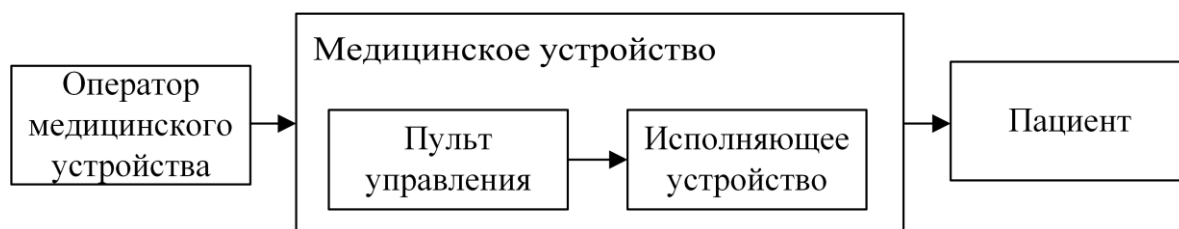


Рисунок 1 – Схема взаимодействия «Оператор медицинского устройства – медицинское устройство – пациент»

Целесообразно разделить медицинское устройство на две составляющих - пульт управления и удаленное исполняющее устройство. Пульт управления отвечает за установку параметров физического поля, генерируемого исполняющим устройством, которое взаимодействует непосредственно с пациентом. Такое решение предоставляет различные преимущества, например: повышение мобильности пациента, за счёт отсутствия прямого взаимодействия между оператором устройства (врачом) и исполняющим устройством [4], в то же время, это позволяет врачу избежать влияния физических полей данного медицинского прибора.

Схему взаимодействия «Оператор – пульт управления – исполняющее устройство – пациент» возможно преобразовать к виду, когда с одним пультом управления одновременно работает множество исполняющих устройств.

Целью нашей работы являлась разработка программного обеспечения на базе операционной системы Android для пульта управления несколькими медицинскими устройствами.

Основная часть

Следуя идеологии "один пульт управления - множеству исполняющих устройств", схема взаимодействия оператора и пациента примет вид, представленный на рисунке 2.

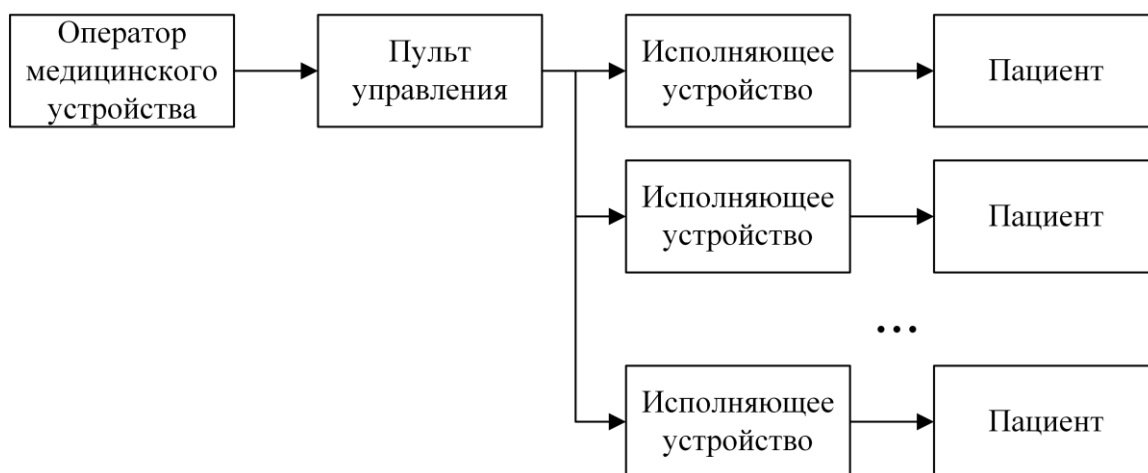


Рисунок 2 – Схема взаимодействия одного пульта управления с несколькими исполняющими устройствами

Современные средства коммуникации, используемые для подключения физиотерапевтических устройств, а также аппаратные средства ЭВМ позволяют создавать сложные структуры подключения приборов. В лечебной практике полезной будет модель,

при которой к одному пульту управления подключается множество исполняющих устройств в соответствии с топологией «Звезда». Такой способ организации взаимодействия позволяет сократить количество пультов управления, а, следовательно, и операторов, задействованных в лечебном процессе. Данный факт существенно снижает затраты на выполнение одной лечебной процедуры.

Необходимо заметить, что разрабатывать специализированные пульты управления каждому физиотерапевтическому устройству, стоит больших материальных затрат [2]. Следовательно, выгоднее в качестве пульта использовать готовые платформы, такие как, мобильное устройство или персональный компьютер. Наиболее распространённым решением, является разработка пультов управления физиотерапевтическими устройствами на базе персонального компьютера под управлением операционной системы Windows.

Целевой платформой, в качестве пульта управления, были выбраны мобильные устройства на базе операционной системы Android. Их преимущества рассмотрены в [3, 7].

Протокол взаимодействия

С целью унификации разрабатываемого пульта управления физиотерапевтическими устройствами целесообразно представить состояния этих устройств с помощью набора целочисленных характеристик. При этом каждая характеристика имеет атрибуты, такие как возможность записи и возможность чтения. Характеристики с атрибутом «для чтения» представляют состояния устройств, обусловленные внешними факторами среды, например, работа от внешнего источника питания или встроенной аккумуляторной батареи, напряжение аккумулятора и так далее. Характеристики с атрибутом «для записи», представляют параметры приборов, отвечающие за генерацию физических воздействий, например, такие как амплитуда, частота, длительность импульса, мощность, давление, время до выключения прибора и так далее.

При применении такой методологии, чтобы получить текущие состояния устройств, необходимо считать состояния приборов и отслеживать изменения их характеристик, для этого необходимо организовать бесконечный цикл опроса состояния устройств. Изменение состояния устройства требует записи соответствующей характеристики.

Архитектура

В связи с тем, что графический интерфейс пользователя (ГИП) достаточно сложный целесообразно использовать шаблоны проектирования программного обеспечения, также называемые паттернами. Для реализации сложных ГИП в настоящее время рекомендуется применять паттерн Model-View-ViewModel (MVVM), позволяющий разделить модель и ее представление друг от друга, в соответствии с рисунком 3. Применение паттерна MVVM тесно связано с применением паттерна Наблюдатель [6].

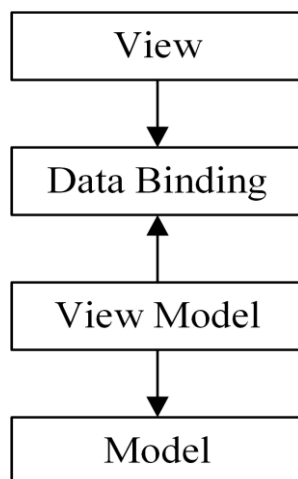


Рисунок 3 - MVVM

Диаграмма взаимодействия ключевых классов приложения, ориентированного на работу со многими физиотерапевтическими устройствами приведена на рисунке 4.

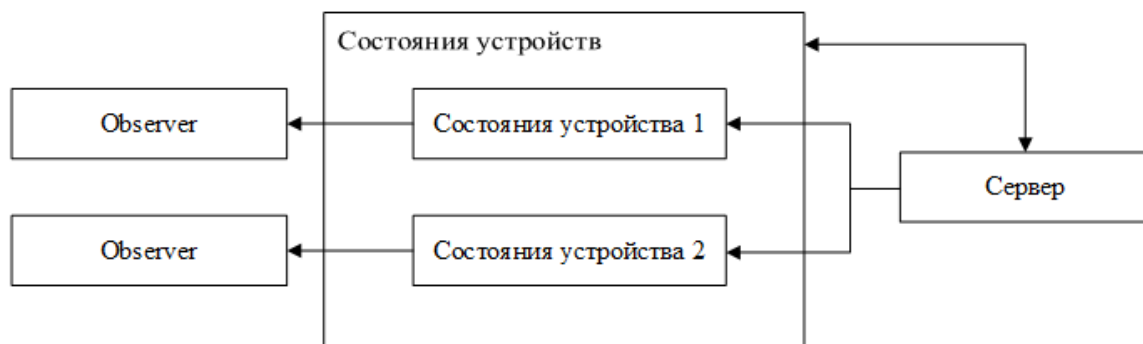


Рисунок 4 – Диаграмма связей наблюдателей и наблюдаемых состояний

Главным элементом программы управления множеством физиотерапевтических устройств является список состояний устройств. Данный список хранит сущности всех состояний устройств. Каждая сущность состояния устройства хранит значения всех характеристик, считанных с физиотерапевтического устройства, а также целевые характеристики, установленные оператором посредством ГИП. Каждая сущность состояния устройства реализует интерфейс `IObservable`, входящий в состав паттерна Наблюдатель, и автоматически оповещает сущности `Observer` при внесении изменений состояние устройства. Каждый объект `Observer` реализует интерфейс `IObserver`, входящий в состав паттерна Наблюдатель и отвечает за перерисовку ГИП соответствующего физиотерапевтического устройства.

Синхронная природа большинства драйверов устройств коммуникационных протоколов требует создания сущности, отвечающей за циклический опрос всех состояний каждого устройства в строгой последовательности с целью синхронизации состояний управляющих устройств с физиотерапевтическими устройствами. Для решения данной задачи была создана сущность `Сервер`, которая имеет информацию обо всех состояниях подключенных физиотерапевтических устройств. `Сервер`, находящийся в бесконечном цикле в отдельном потоке приложения, поочередно обрабатывает состояния устройств при помощи следующих шагов:

- Запрос с физиотерапевтического устройства всех характеристик, предназначенных для чтения.

– Для каждой характеристики, предназначенной для записи, сравнение текущего значения этой характеристики, хранимой на физиотерапевтическом устройстве с соответствующей целевой характеристикой. Если две эти характеристики имеют различные значения, то производится запись значения целевой характеристики в соответствующую характеристику для установки параметра физиотерапевтического устройства.

За счет реализации паттерна Наблюдатель сущности Observer получают события об изменениях соответствующего состояния устройства, при получении которого объект Observer инициирует перерисовывание соответствующих областей ГИП актуализированной информацией.

Работа с физиотерапевтическими устройствами осуществляется в одном рабочем окне программы, которое представлено формой типа Master-Details. Форма состоит из трёх деятельностей, так называемых Activity [1] – «Выбор устройства», «Выбор подключенного устройства» и «Работа с устройством». Такая организация позволяет одновременно контролировать состояния всех устройств, но при этом не перегружает экран излишней информацией, что облегчает работу оператора.

Пользовательский интерфейс

При запуске приложения инициируется окно, с кнопкой «Обновить список устройств». По нажатию этой кнопки, программа осуществляет анализ всех коммуникационных интерфейсов, например, таких как Ethernet, USB, RS232 и выполняет поиск подключенных физиотерапевтических устройств. Список доступных для подключения устройств, отображается в левом нижнем углу окна приложения, в деятельности «Выбор устройства», такая ситуация представлена на рисунке 5. Выбирая какой-либо прибор, запускается процедура «Подключение», при выполнении которой программа создает новую сущность состояния устройства, ассоциированную с подключаемым устройством, и добавляет созданную сущность в перечень подключенных устройств. Вместе с созданием сущности состояния устройства создается ассоциированная сущность Observer, отвечающая за взаимодействие с ГИП. На текущей итерации опроса устройств Сервер актуализирует свойства состояния устройства, которые затем при помощи соответствующего Observer'a обновляются.

При выборе устройства в списке подключенных приборов, инициализируется деятельность «Работа с устройством», в которой возможна манипуляция параметрами этого медицинского прибора. Пример ГИП с инициализированной деятельностью «Работа с устройством» приведен на рисунке 6.

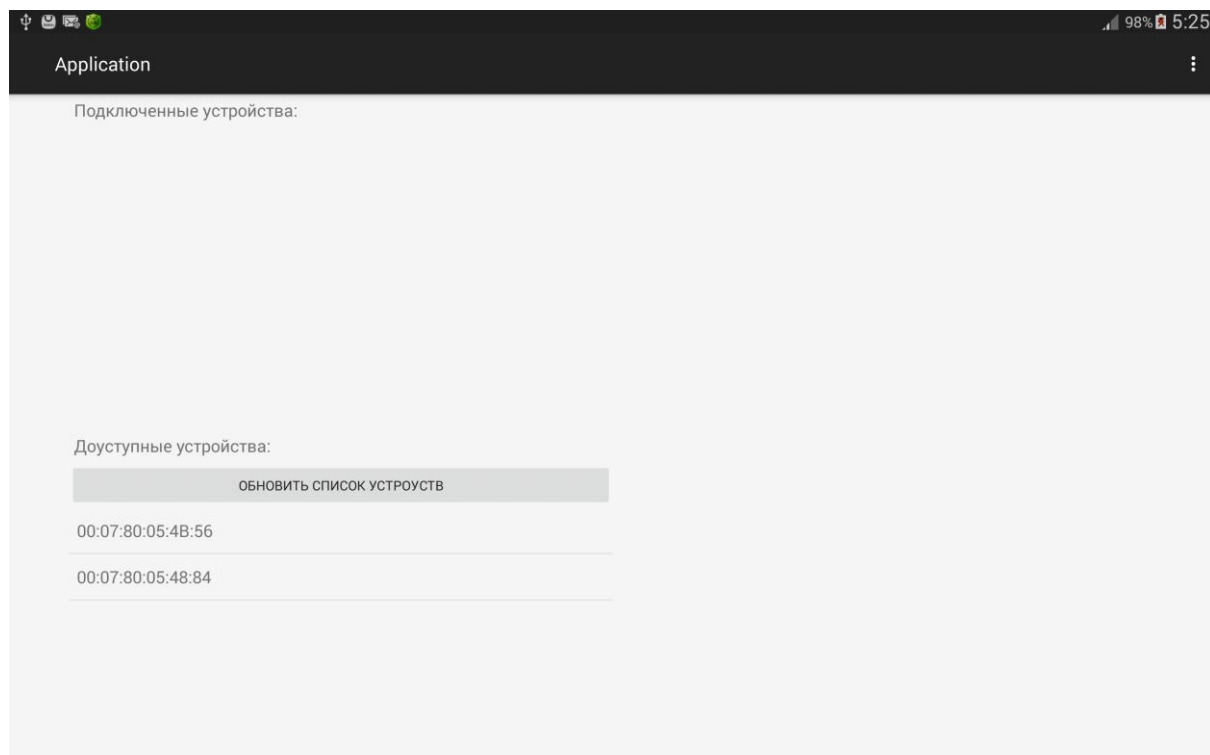


Рисунок 5 — Интерфейс приложения. Поиск и отображение доступных для подключения устройств

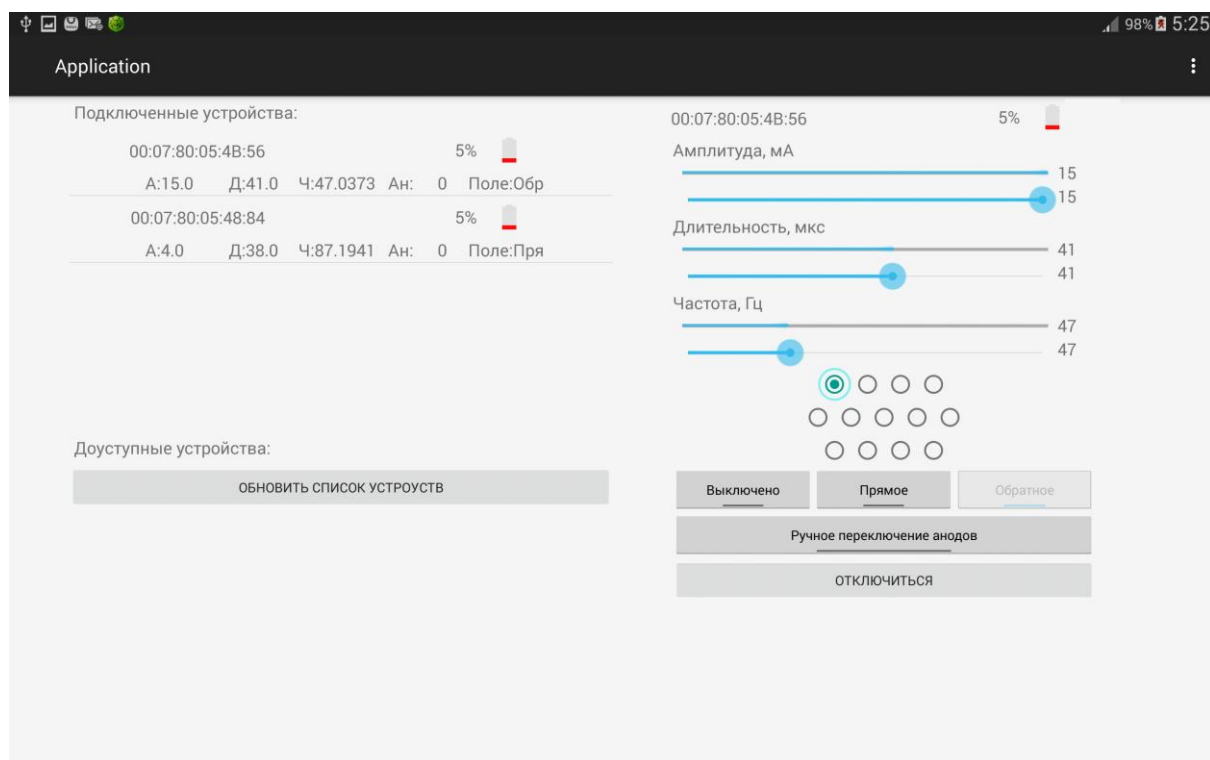


Рисунок 6 — Интерфейс. Манипуляция параметрами

Выводы

Результатом работы является разработанное приложение, которое служит примером реализации программного обеспечения для пульта управления несколькими медицинскими физиотерапевтическими устройствами на базе операционной системы Android. Благодаря использованию системы сервисов существует возможность управлять различными устройствами без необходимости изменения исходного кода программы.

Список литературы

1. Android Developer [электронный ресурс] / Develop API Guides/ Activities : [сайт]. URL: <http://developer.android.com/guide/components/activities.html>
2. Benjamin A. Suter, Timothy O'Connor, Vijay Iyer, Leopoldo T. Petreanu, Bryan M. Hooks, Taro Kiritani, Karel Svoboda and Gordon M. G. Shepherd. Ephus: multipurpose data acquisition software for neuroscience experiments // «Frontiers in neural circuits» - 2010. – Volume 4 – article 100.
3. Karim Yaghmour, Embedded Android // «O'Reilly Media» – 2013. – 412 p.
4. Nuno Costa, Tiago Araujo, Neuza Nunes, Hugo Gamboa, Multi-purpose Electrostimulator Software // «Scitepress» - 2013.
5. JavaWorld/ Observer and observable – URL: <http://www.javaworld.com/article/2077258/learn-java/observer-and-observable.html>
6. CodeProject [электронный ресурс] // Model-View-ViewModel : [сайт]. URL: <http://www.codeproject.com/Articles/100175/Model-View-ViewModel-MVVM-Explained>
7. М.В.Бабич, А.В. Фрелих, Д.Г. Шемякин, Е.А. Арсентьев. Разработка многоцелевого программного обеспечения для управления медицинскими приборами на базе операционной системы Android// «Интернаука» - 2015. - сборник 27 - 105 с.

References

1. Android Developer / Develop API Guides/ Activities — URL: <http://developer.android.com/guide/components/activities.html>
2. Benjamin A. Suter, Timothy O'Connor, Vijay Iyer, Leopoldo T. Petreanu, Bryan M. Hooks, Taro Kiritani, Karel Svoboda and Gordon M. G. Shepherd. Ephus: multipurpose data acquisition software for neuroscience experiments // «Frontiers in neural circuits» - 2010. – Volume 4 – article 100.
3. Karim Yaghmour, Embedded Android // «O'Reilly Media» – 2013. – 412 p.
4. Nuno Costa, Tiago Araujo, Neuza Nunes, Hugo Gamboa, Multi-purpose Electrostimulator Software // «Scitepress» - 2013.
5. JavaWorld/ Observer and observable – URL: <http://www.javaworld.com/article/2077258/learn-java/observer-and-observable.html>
6. CodeProject/Model-View-ViewModel - URL: <http://www.codeproject.com/Articles/100175/Model-View-ViewModel-MVVM-Explained>
7. М.В. Babich, A.V. Frelich, D.G.Shemyakin, E.A. Arsentiev. Development of multi-purpose software for management medical devices using operation system Android. // «Internauka» - 2015. - Volume 27 - 105 p.